WE CLAIM:

1. A data processing apparatus for performing a data processing operation on first and second floating point data elements, the first floating point data element specifying a first exponent and the second floating point data element specifying a second exponent, the data processing apparatus comprising:

processing logic providing multiple processing paths which are selectable to perform the data processing operation, including a first processing path operable to perform the data processing operation if a predetermined alignment condition exists;

at least one detector logic unit operable to receive both said first exponent and said second exponent, and to detect the presence of said predetermined alignment condition, each detector logic unit comprising:

half adder logic operable to perform a number of half adder operations to logically subtract one of the first and second exponents from the other of the first and second exponents to produce at least a sum data value of sum and carry data values representing the result of the number of half adder operations; and

generation logic operable to receive the sum data value and to generate a select signal which is set if the sum data value has a predetermined value indicating the existence of said predetermined alignment condition;

the processing logic being operable to select the first data processing path to perform the data processing operation if the select signal from one of said at least one detector logic units is set.

2. A data processing apparatus as claimed in Claim 1, wherein the number of half adder operations performed by the half adder logic is a plurality of half adder operations.

3. A data processing apparatus as claimed in Claim 1, wherein if the select signal from one of said at least one detector logic units is set, the processing logic is operable to prevent performance of the data processing operation in the processing paths other than the first processing path.

4. A data processing apparatus as claimed in Claim 1, wherein the predetermined alignment condition specifies that the first and second floating point data elements require at most a one-bit alignment, and the at least one detector logic unit is operable to detect whether the first and second exponents differ by one by determining whether the sum value has a value of -2, if the sum value has a value of -2 the at least one detector logic unit being operable to generate a shift signal in addition to the select signal.

5. A data processing apparatus as claimed in Claim 4, wherein the at least one detector logic unit is operable to detect whether the first and second exponents are equal or differ by one by determining whether the sum value has a value of -1 or -2, if the sum value has a value of -1 or -2 the at least one detector logic unit being operable to generate the select signal.

6. A data processing apparatus as claimed in Claim 4, wherein both the first and second exponents have n bits, and the half adder logic comprises:

first n-bit half adder logic operable to perform a first half adder operation to logically subtract said one exponent from said other exponent to produce an intermediate sum value and an intermediate carry value; and

additional logic operable to perform at least a partial second half adder operation to logically add the intermediate sum value and intermediate carry value to generate said sum value.

7. A data processing apparatus as claimed in Claim 6, wherein the additional logic comprises XOR logic operable to perform an XOR operation on corresponding bits of the intermediate sum value and the intermediate carry value other than the least significant bit.

8. A data processing apparatus as claimed in Claim 4, wherein the processing logic has a plurality of pipeline stages, and each processing path comprises multiple pipeline stages, the at least one detector logic unit being operable to generate the shift signal for input to a first pipeline stage of the first processing path, this first pipeline stage

containing shift logic, and the shift signal being used to control the operation of the shift logic.

9. A data processing apparatus as claimed in Claim 8, wherein the first floating point data element specifies a first mantissa and the second floating point data element specifies a second mantissa, and the shift logic comprises first shift logic provided to selectively perform a shift operation on the first mantissa and second shift logic provided to selectively perform a shift operation on the second mantissa, the at least one detector logic unit comprising a first detector logic unit associated with the first shift logic and a second detector logic unit associated with the second shift logic, the half adder logic of the first detector logic unit being operable to logically subtract the first exponent from the second exponent and the half adder logic of the second detector logic unit being operable to logically subtract the second exponent from the first exponent.

10. A data processing apparatus as claimed in Claim 8, wherein the first pipeline stage is common to the multiple processing paths.

11. A data processing apparatus as claimed in Claim 1, wherein the data processing operation is an unlike-signed addition operation.

12. A method of determining a processing path of a data processing apparatus to perform a data processing operation on first and second floating point data elements, the first floating point data element specifying a first exponent and the second floating point data element specifying a second exponent, the data processing apparatus having processing logic providing multiple processing paths which are selectable to perform the data processing operation, including a first processing path operable to perform the data processing operation if a predetermined alignment condition exists, the method comprising the steps of:

(a) providing at least one detector logic unit which receive both said first exponent and said second exponent, and within each detector logic unit detecting the presence of said predetermined alignment condition by performing the steps of:

(a)(i) employing half adder logic to perform a number of half adder operations to logically subtract one of the first and second exponents from the other of the first and second exponents to produce at least a sum data value of sum and carry data values representing the result of the number of half adder operations; and

(a)(ii) generating a select signal which is set if the sum data value has a predetermined value indicating the existence of said predetermined alignment condition;

(b) selecting the first data processing path to perform the data processing operation if the select signal from one of said at least one detector logic units is set.

13. A method as claimed in Claim 12, wherein the number of half adder operations performed by the half adder logic is a plurality of half adder operations.

14. A method as claimed in Claim 12, wherein if the select signal from one of said at least one detector logic units is set, the method further comprises the step of preventing performance of the data processing operation in the processing paths other than the first processing path.

15. A method as claimed in Claim 12, wherein the predetermined alignment condition specifies that the first and second floating point data elements require at most a one-bit alignment, and the at least one detector logic unit detects at said step (a) whether the first and second exponents differ by one by determining whether the sum value has a value of -2, if the sum value has a value of -2 the at least one detector logic unit generating a shift signal in addition to the select signal.

16. A method as claimed in Claim 15, wherein the at least one detector logic unit detects at said step (a) whether the first and second exponents are equal or differ by one by determining whether the sum value has a value of -1 or -2, if the sum value has a value of -1 or -2 the at least one detector logic unit generating the select signal.

17. A method as claimed in Claim 15, wherein both the first and second exponents have n bits, and said step (a)(i) comprises the steps of:

performing a first half adder operation to logically subtract said one exponent from said other exponent to produce an intermediate sum value and an intermediate carry value; and

performing at least a partial second half adder operation to logically add the intermediate sum value and intermediate carry value to generate said sum value.

18. A method as claimed in Claim 17, wherein the step of performing at least a partial second half adder operation comprises the step of performing an XOR operation on corresponding bits of the intermediate sum value and the intermediate carry value other than the least significant bit.

19. A method as claimed in Claim 15, wherein the processing logic has a plurality of pipeline stages, and each processing path comprises multiple pipeline stages, the at least one detector logic unit generating the shift signal for input to a first pipeline stage of the first processing path, this first pipeline stage containing shift logic, and the shift signal controlling the operation of the shift logic.

20. A method as claimed in Claim 19, wherein the first floating point data element specifies a first mantissa and the second floating point data element specifies a second mantissa, and the shift logic comprises first shift logic provided to selectively perform a shift operation on the first mantissa and second shift logic provided to selectively perform a shift operation on the second mantissa, the at least one detector logic unit comprising a first detector logic unit associated with the first shift logic and a second detector logic unit associated with the second shift logic, at said step (a) the half adder logic of the first detector logic unit logically subtracting the first exponent from the second exponent and the half adder logic of the second detector logic unit logically subtracting the second exponent from the first exponent.

21. A method as claimed in Claim 19, wherein the first pipeline stage is common to the multiple processing paths.

22. A method as claimed in Claim 12, wherein the data processing operation is an unlike-signed addition operation.